

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

**Mechanical Design Assembly Translation  
Including Preservation Of Assembly Constraints/Associativity**

Inventor(s):

**Sunderarajan G. Karaikurichi**

Prepared by:

Columbia IP Law Group, LLC  
4900 S.W. Meadows Rd., Suite 109  
Lake Oswego, Oregon 97035  
(503) 534-2800

"Express Mail" label number EL743034062US

**Mechanical Design Assembly Translation Including Preservation Of Assembly**

**Constraints/Associativity**

**BACKGROUND OF THE INVENTION**

5

1. Field of the Invention

The present invention relates to the field of computer aided design (CAD). More specifically, the present invention relates to methods and apparatuses for translating mechanical design assemblies (including their constraints or associativity) from one representation to another.

10

2. Background Information

With the advance of computing technology, mechanical designers have long since turned to computer-aided design (CAD) software to assist them in designing ever more complex mechanical designs. To-date, numerous CAD software are available from different vendors. Examples of these CAD software include but are not limited to SolidWorks ProEngineer and Mechanical Desktop available from Autodesk, Inc. of San Rafael of CA, assignee of the present invention.

15

With the proliferation of the different CAD software available in the marketplace, mechanical designers (typically of different organizations) often find themselves having to translate the mechanical designs of each other, as they collaborate and share their designs. The reason being, even though fundamentally all CAD software use geometric primitives and solid modeling (in the case of 3D parts) to describe mechanical parts, but different vendors employ different data formats and/or organizations, as well as different modeling approaches. That is, different CAD software will model a geometry, such as a line, with different number of geometric primitives of different kinds, or a solid, such as a 3D cylinder, with

20

25

different solid models. As a result, the process of translation involves not only conversion from one data format/organization to another, but also from one modeling approach to another. Thus, in the translation of a 2D geometry, such as a line, more or less geometric primitives (potentially of different kinds) may result, and  
5 in the translation of a 3D solid, typically, a solid model is approximated by a collection of surface geometric entities instead.

As a result of the ever increasing need for mechanical designers to collaborate and share their designs, numerous data interchange formats and part translation techniques, tools and utilities are known in the art. However, in the real  
10 world, increasingly mechanical designers are working with assemblies that are made up of ever increasing number of parts. Further, these assemblies have assembly constraints or associativity (hereinafter, simply constraints), such as one sub-assembly or part is to “mate” with another in a particular manner, other sub-assemblies or parts are to be “flushed” with each other, and so forth, as the sub-  
15 assemblies and parts are joined together to form the assemblies. Thus, having only parts translation is no longer sufficient for mechanical designers dealing with complex assemblies having a large number of sub-assemblies and/or parts, as well as assembly constraints. What is needed is an effective approach to assist mechanical designers in translating assemblies in substantially their entirety.

20

## SUMMARY OF THE INVENTION

A constraint translator first determine geometry entities within a number of  
5 translated representations of sub-assemblies/parts of a mechanical design  
assembly corresponding to geometric entities within a number of pre-translation  
representations of the sub-assemblies/parts of the mechanical design assembly,  
constrained by one or more assembly constraints of the mechanical design  
assembly. The constraint translator then correspondingly constrains the determined  
10 counterpart sub-assemblies/parts of which the determined geometric entities within  
the translated representations are part of to effectively translate the one or more  
assembly constraints of the mechanical design assembly.

In one embodiment, the assembly constraint translator is provided as a  
component of an assembly translator. The assembly translator first invokes a part  
15 translator to translate the parts of the sub-assemblies of a mechanical design  
assembly, and thereafter invokes the assembly constraint translator to translate the  
assembly constraints as summarized above.

In one embodiment, the assembly translator, together with the part and  
constraint translators are provided as an integral part of a computer-aided  
20 mechanical design software.

## BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references  
5 denote similar elements, and in which:

**Figure 1** illustrates a CAD software incorporated with the teachings of the present invention, in accordance with one embodiment;

**Figure 2** illustrates the relevant operational flow of the main routine of **Fig. 1**, in accordance with one embodiment;

10 **Figures 3a-3b** illustrate the relevant operational flow of the assembly translator of **Fig. 1**, in accordance with one embodiment;

**Figures 4a-4b** illustrate an example hierarchical representation of an example assembly, and an example data structure for describing the example hierarchical representation of the example assembly, in accordance with one  
15 embodiment;

**Figure 5** illustrates an example data structure for tracking the correspondence between the pre-translation and translated representations of the sub-assemblies and parts, in accordance with one embodiment;

**Figure 6** illustrates the relevant operational flow of the constraint translator of  
20 **Fig. 1**, in accordance with one embodiment; and

**Figure 7** illustrates one embodiment of an example computer system suitable for programming with instructions implementing the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

In the following description, various aspects of the present invention will be described. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all aspects of the present invention. For purposes of explanation, specific numbers, materials and configurations are set forth in order to provide a thorough understanding of the present invention. However, it will also be apparent to one skilled in the art that the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

Parts of the description will be presented in terms of operations performed by a computer system, using terms such as data, data structures, determining, constraining, and the like, consistent with the manner commonly employed by those skilled in the art to convey the substance of their work to others skilled in the art. As well understood by those skilled in the art, the quantities of these operations and the operations themselves take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, and otherwise manipulated through mechanical and electrical components of the computer system. Moreover, the term computer system as used herein includes general purpose as well as special purpose data processing machines, systems, and the like, that are standalone, adjunct or embedded.

Various operations will be described as multiple discrete steps in turn, in a manner that is most helpful in understanding the present invention, however, the order of description should not be construed as to imply that these operations are necessarily order dependent. In particular, these operations need not be performed in the order of their presentation. The phrase "in one embodiment" will be employed

from time to time, and it is not intended to necessarily refer to the same embodiment, although it may.

Referring now **Figure 1**, wherein a block diagram illustrating an example CAD software incorporated with teachings of the present invention is shown. As illustrated, CAD software **100** is advantageously provided with constraint translator **118** for translating constraints of a mechanical design assembly having a number of sub-assemblies and/or parts, and one or more assembly constraints. For the illustrated embodiment, constraint translator **118** is advantageously provided in conjunction with hierarchy translator **117**, as components of assembly translator **114**, which itself is provided in conjunction with main **112** and parts translator **116**, forming design translator **110**. Together, the elements of design translator **110** cooperate with each other to effectuate the desired automated translation of a mechanical design assembly having a number of sub-assemblies and/or parts, and one or more assembly constraints.

Except for the incorporation of constraint translator **118**, in conjunction with hierarchy translator **114**, forming assembly translator **114**, and in conjunction with main **112** and parts translator **116**, forming design translator **110**, CAD software **100** including part translator **116** are intended to represent a broad range of these elements known in the art. Their constitutions and operations are known in the art, and will not be otherwise further described. Main **112**, assembly translator **114**, including hierarchy translator **117** and constraint translator **118**, will be described in more details in turn below.

**Figure 2** is a flow chart illustrating the relevant aspects of the operational flow of main **112**, in accordance with one embodiment. As illustrated, main **112** is provided primarily to allow design translator **110** to be advantageously used for

conventional translation of parts, as well as for the novel automated translation of a mechanical design assembly having a number of sub-assemblies and/or parts, and one or more assembly constraints. Upon receipt of a request for translation, main **112** determines if the request is for the translation of a part or the translation of an assembly, block **202**, and invokes parts translator **116** or assembly translator **114** accordingly, block **204** or **206**. Conveyance of the request, and denoting of the request type, may be made in any one of a number of techniques known in the art, including but are not limited to messaging, event notification, parameter passing, control register setting, and the like.

**Figure 3** is a flow chart illustrating the relevant aspects of the operational flow of assembly translator **112**, more specifically, hierarchy translator **117**. As illustrated, upon invocation, hierarchy translator **117**, selects a sub-assembly branch of the mechanical design assembly (e.g. the sub-assembly branch originating from sub-assembly **404a** or sub-assembly **404b** of example mechanical design assembly **400** of **Fig. 4**), block **302**. It is unimportant which sub-assembly branch get selected first. In one embodiment, hierarchy translator **117** follows a pre-determined “left-to-right” order, in another, hierarchy translator **117** follows a pre-determined “right-to-left” order. [Those skilled in the art will appreciate that “left-to-right” and “right-to-left” are merely shorthand references for ease of understanding. During operation, hierarchy translator **117** basically processes the different corresponding “segments” of the data structure iteratively, “segment” by “segment”. See e.g. example data structure **410** of **Fig. 4b** having data “segments” **412**.].

At block **304**, hierarchy translator **117** selects the “next” sub-assembly/part of the selected sub-assembly branch (e.g. sub-assembly **404a**, if hierarchy translator **117** is processing from “left-to-right”), records its spatial position in the mechanical design assembly. At block **306**, hierarchy translator **117** determines if the selected



“next” sub-assembly/part is a sub-assembly or a part. If the selected “next” sub-assembly/part is a sub-assembly, the process continues at block **304** and then block **306** again. That is, hierarchy translator **117** selects the “next” sub-assembly/part of the selected sub-assembly branch, records its spatial position in the mechanical design assembly, and determines if the selected “next” sub-assembly/part is a sub-assembly or a part. Eventually, hierarchy translator **117** will determine at block **306** that the selected “next” sub-assembly/part is a part. The process then continues at block **310**.

At block **310**, hierarchy translator **117** invokes parts translator **114** to translate the selected part. As alluded to earlier, translation not only includes conversion of the data format and/or organization being used, but typically also includes a change in the modeling approach being used. In particular, for 3D objects, most likely, solid representations will be generated. At block **312**, hierarchy translator **117** determines if more parts are to be translated for this particular “location” of the assembly hierarchy. If more parts are to be translated, the process continues at block **310**, otherwise the process continues at block **312**, where hierarchy translator **117** “moves up” one hierarchical level of the assembly hierarchy, and determines if there are additional “sub-assembly” branch segments to be analyzed, block **314**. If one or more “sub-assembly” branch segments are still to be analyzed, the process continues back at block **302**, and if all “sub-assembly” branch segments have been processed, the process continues at block **318**.

At block **318**, hierarchy translator **117** invokes constraint translator **118** to translate the assembly constraints of the mechanical design assembly. Hierarchy translator **117** repeats block **318** as long as there are assembly constraints to be translated, block **320**.

As alluded to earlier, **Figure 4a** illustrates an example hierarchical representation **402** of an example mechanical design having a number of sub-

assemblies **404a-404c**, and some of the sub-assemblies, e.g. sub-assembly **404a**, having additional sub-assemblies, e.g. sub-assembly **404b**, while others include one or more parts, e.g. parts **406a-406c**. Note that hierarchy **402** represents the assembly's sub-assemblies and parts hierarchical relationship in its pre-translation as well as post-translation state. In other words, while translation alters the data format and/or organization, , translation does not fundamentally alter the number of "entities" employed to model an assembly, and the hierarchical relationship between the existing sub-assemblies and parts.

**Figure 4b** illustrates an example corresponding data structure **410** having a number of data segments **412** suitable for use to describe an assembly hierarchy, in accordance with one embodiment. For the illustrated embodiment, each data segment **412** includes an assembly, sub-assembly or part identifier **414**, **420** or **426**. Additionally, for the sub-assembly and part data segments **412**, each data segment **412** further includes a spatial location **422** or **428** of the sub-assembly/part in the assembly. For the assembly and sub-assembly data segments **412**, each data segment **412** also includes one or more pointers **418** pointing to the immediately constituting sub-assemblies or parts. For the parts data segments **412**, each data segment **412** also includes data describing the parts or parts data **430**.

Example data structure **410** may be used to store the sub-assembly and parts data of the assembly in their pre-translation or post-translation state.

For the illustrated embodiment, an example mapping data structure, such as table **500** of **Fig. 5** having a number of mapping entries such as entries **502** is used to track the correspondence between the pre-translation and post-translation representations of the sub-assemblies and parts. Note that while the representations, such as parts data **430** may be different before and after translation, nevertheless, there is a one-to-one correspondence between a pre-

translation representation and a post-translation representation of an existing sub-assembly/part.

**Figure 6** illustrates the relevant operation flow of constraint translator **118** for translating assembly constraints of the assembly, in accordance with one

5 embodiment. For each assembly constraint, the process starts at block **602** where constraint translator **118** first identifies the “involved” sub-assemblies/parts, that is, the sub-assemblies/parts constrained by the particular assembly constraint. As described earlier, examples of assembly constraints include but are not limited to “mate”, “flush”, “angle” and “rotation” constraints. At block **604**, constraint translator  
10 **118** determines, for each “involved” sub-assembly/part, the geometry elements (or entities) in the pre-translation representation that are actually constrained by the particular assembly constraint. At block **606**, constraint translator **118** determines, for each of the actually constrained geometry elements (or entities), a number of sampling points, and their coordinates. Sampling points may be selected based on  
15 a number of empirically pre-determined approaches. For example, the sampling points may be selected by always including the vertices and the centroid of a geometry element. Additionally, the number of sampling points may be employed to ensure certain sampling point density is achieved. The additional sampling points to increase the sampling point density may be the additional centroids of the different  
20 systematically partitioned areas of a geometry element. Determination of the vertices and the centroid of a geometry element as well as the partitioning of a geometry element are known in the art, accordingly will not be further described.

At block **608**, constraint translator **118** determines, for each of the actually constrained geometry elements (or entities), their corresponding geometry element  
25 or elements (or entity/entities), using the selected sampling points, more specifically the coordinates of the selected sampling points. At block **610**, upon identifying the corresponding geometry element or elements (or entity/entities), constraint

translator **118** identifies the parts in the translated representation modeled (in whole or in part) by the identified constrained geometry elements of the translated representation. At block **612**, if applicable, constraint translator **118** further identifies the sub-assemblies of the translated representation to which the identified parts of the translated representation are members of. At block **614**, constraint translator **118** applies the constraints to the identify geometry element/elements (entity/entities) of the corresponding sub-assemblies/parts, effectively translating the assembly constraint.

Process **600** is repeated by constraint translator **118** for each assembly constraint.

**Figure 7** illustrates one embodiment of a computer system suitable to be programmed with programming instructions implementing the CAD software incorporated with the constraint translator and other aspects of the present invention. As illustrated, computer system **700** includes one or more processors **702** and system memory **704**. Additionally, computer system **700** includes mass storage devices **706** (such as diskette, hard drive, CDROM and so forth), input/output devices **708** (such as keyboard, cursor control and so forth) and communication interfaces **710** (such as network interface cards, modems and so forth). The elements are coupled to each other via system bus **712**, which represents one or more buses. In the case of multiple buses, they are bridged by one or more bus bridges (not shown). Each of these elements perform its conventional functions known in the art. In particular, system memory **704** and mass storage **706** are employed to store a working copy and a permanent copy of the programming instructions implementing the teachings of the present invention. The permanent copy of the programming instructions may be loaded into mass storage **706** in the factory, or in the field, as described earlier, through a distribution

medium (not shown) or through communication interface **710** (from a distribution server (not shown)). The constitution of these elements **702-712** are known, and accordingly will not be further described.

- 5            Thus, a method and an apparatus for translating a mechanical design assembly including its assembly constraints in an automated manner have been described. Those skilled in the art will recognize that the present invention is not limited by the details described, instead, the present invention can be practiced with modifications and alterations within the spirit and scope of the appended claims.
- 10          The description is thus to be regarded as illustrative instead of restrictive on the present invention.